AMENDMENTS TO THE SPECIFICATION

1) Please replace paragraph [0002] with the following amended version.

[0002] This application relates to U.S. Application 10/017,742 (Attorney Docket 01341) ~~Attorney Docket No. 36968-265386 (BS01341)~~, filed by Matz et al. on December 14, 2001, entitled "System and Method for Utilizing Television Viewing Patterns," which is incorporated herein by reference. This application also relates to U.S. Application 11/636,068 (Attorney Docket 01342), ~~Attorney Docket No. 36968-265387 (BS01342)~~ filed by Matz et al. on December 14, 2001, entitled "System and Method for Identifying Desirable Subscribers," now issued as U.S. Patent 7,212,979, which is incorporated herein by reference. This application also relates to U.S. Application 09/496,825 ~~Serial No. 09/496825~~, filed February 1, 2000, now issued as U.S. Patent 6,983,478 (Attorney Docket 95003 CON), which is incorporated herein by reference.

2) Please replace paragraph [0039] with the following amended version.

[0039] The systems and methods of the present invention may be advantageously implemented with the systems and methods disclosed in U.S. Application 10/017,742, ~~a patent application~~ filed by Matz et al. on December 14, 2001, entitled "System and Method for Utilizing Television Viewing Patterns," (Attorney Docket No. 36968-265386 (BS01341)) which is incorporated herein by reference. Under embodiments disclosed therein, a subscriber's television viewing patterns are combined with programming and advertising media-content detail to determine the subscriber's content choices.

3) Please insert the following paragraphs after paragraph [0074]:

[0074A] Exemplary embodiments use a collector, associated with a subscriber's set top box ("STB"), to obtain data about any "events" -- subscriber actions or changes in programming -- that are of interest. Data about virtually any events, from channels watched to volume changes to interactive applications invoked, may be captured with the collector. Event records comprising such data, as well as the identity of the application involved and the event time, are buffered. Periodically or on command, event records are uploaded from the buffer to a merge processor such as through an interactive network that allows for duplex communication with the STB. The merge processor, which may be a head end server or a workstation computer forming part of or coupled to the media delivery network, receives (1) the event data and (2) content data that identifies programming content broadcast or delivered throughout the region in which the system is deployed. Timelines showing particular events over time may then be generated for each subscriber. Rather than just determining the channel viewed and time of day, the event timelines describe the programming or interactive applications selected by or shown to a subscriber over a selected period of time (*e.g.*, 24 hours).

[0074B] **A c**lickstream processor collects information to create a "journal" or log about all events or selected events of interest. An event is an action or a change in the state of a STB that is deemed important to building a knowledge base on subscribers or their viewing patterns. For example, an event can include key presses to change channels or volume, mute, to enter the navigator for the interactive system, to turn the STB off or on, to fast forward, to pause or to rewind a video obtained via the video on demand application. The events include applications called by the subscriber, such as interactive gaming applications, an electronic program guide, a video on demand or near video on demand application, a home-shopping application or a particular company's interactive application, such as The Weather Channel's weather on demand, World Span's travel on demand or Light Span's educational interactive application. Events

include subscriber use of and control commands to peripheral devices coupled to the STB or a subscriber's display device, such as a VCR or videodisk player.

[0074C] Each application residing on the STB interfaces with the clickstream processor to send selected data for maintaining a desired journal. Assuming that the system is used with an interactive system, many different applications may be deployed over that system and may be triggered by the subscriber. Some fairly typical applications that might be invoked include:

- a cable television application that handles subscriber remote controls (like channel or volume changes);
- an electronic programming guide application such as TV Data, Prevue or Star Sight interactive services;
- an interactive game;
- a video on demand or near video on demand application;
- company specific applications, that might be offered by content provider such as the Weather Channel, MTV, Showtime, etc.; or
- a navigator application to help the user choose options.

Each of these applications, as well as some internal applications that the system 20 may wish to monitor, will be assigned a unique application identifier.

[0074D] Clickstream processor interfaces with the various applications resident in the STB operating system and any third party applications. Note that for systems using other types of STB than the embodiment described in the Figures, those STBs need not have an operating system. Instead, all instructions can be written directly to the memories of those particular STBs. Applications can be added by either downloading entirely new software directly to memory or by downloading new tables as described below.

[0074E] When an application reaches a point where an "event" of interest has been generated, the application stores an event record to memory. The application

then launches to the clickstream kernel the event record, including information such as: (1) the application's identification code (e.g., the "Cable Television Application" or a particular interactive application); (2) a count of the amount of information (number of bytes) to be journaled; (3) a "time stamp" that defines a unique point in time, e.g., by defining the date and time of day, accurate to the hour, minute or second; (4) an identification code for the event, or (5) where the event data was stored. Clickstream kernel uses the information provided by the applications to collect the event data, format it and place it into a buffer. Table I shows the type of information that will be generally sent by the clickstream processor to the buffers.

Table I: Application Event Record	**Size**
Timestamp	6 bytes
Assigned Application ID	16 bits
Number Bytes to Follow (length)	8 bits
Application Specific Data with customized formats and lengths	Multiple Bytes

[0074F] Global table II defines events of interest that each application can identify, collect, store in the "Application Specific Data" field and notify the clickstream kernel. These events could be as simple as a broadcast channel change by pressing the "Chan Up" remote key. All of these event types can be accessed and used by each application. While each application may not use every possible event type, the number of events available for collection allows system to extract any pertinent usage information for analysis. Also, the use of the global table II increases system efficiency because event types can be modified, added or removed.

Table II: EVENT DEFINITIONS

Table II: EVENT DEFINITIONS

Code	Event
Content Related Events	
0x0000	Passive Content Change
Direct Key Presses	
0x0001	TV <> ITV Pressed
0x0002	Power Pressed
0x0003	One (1) Pressed
0x0004	Two (2) Pressed
0x0005	Three (3) Pressed
0x0006	Four (4) Pressed
0x0007	Five (5) Pressed
0x0008	Six (6) Pressed
0x0009	Seven (7) Pressed
0x000A	Eight (8) Pressed
0x000B	Nine (9) Pressed
0x000C	Zero (0) Pressed
0x000D	Channel Up

Code	Event
Application/State Switching Related	
0x0028	AC Power ON
0x0029	Application Switch (Normal)
0x002A	Application Switch (Abnormal)
0x002B	Application Terminated (Normal)
0x002C	Application Terminated (Abnormal)
0x002D	Soft Power OFF
0x002E	Soft Power ON
0x002F	OFF State Polling Event
General	
0x0030	Direct Channel Change
0x0031	Mute
0x0032	Un-Mute
0x0033	Volume Change Below 50%
0x0034	Volume Change Below 25%
0x0035	Volume Change

	Pressed
0x000E	Channel Down Pressed
0x000F	Volume Up Pressed
0x0010	Volume Down Pressed
0x0011	Last Channel Pressed

	Below 10%
0x0036	Volume Change Above 50%
0x0037	Volume Change Above 25%
0x0038	Volume Change Above 10%
0x0039	Change to Interactive Mode
0x003A	Change to Broadcast Mode

[0074G] Note that Table II defines relative volume changes (e.g. "volume change below 50%," "volume change below 25%," etc.). Although the applications could capture the actual key presses that lead to these relative volume changes, that level of detailed information is of little use to system operators. Also, capturing all that detail leads to more records and higher demands upon the transmission network when those records are uploaded. Applications could also be configured to "filter" other unwanted details about other subscriber activities. For example, when subscribers "channel surf" by quickly flipping through a number of channels in a short period of time, the application could be configured not to record channel changes unless the subscriber paused for greater than a certain selected time period (e.g., 15 to 30 seconds). Again, this eliminates information of little use and decreases network traffic.

[0074H] Table III defines a small portion of a sample global channel identification table that proposes codes for identifying national and local broadcasters. Such a table allows any application journaling events which occur while subscribers are viewing broadcast or cable television programs to identify the network carrying the programming content by using a subset of the global

table II. In this way channel lineups can be changed yet the identifier for a broadcast or cable network would stay the same. The use of this mapping scheme eliminates the need to map an ever-changing channel number to a network.

Table III: Broadcast Channel Identification	
0x0100 to	
0x011F	**News/Talk Shows**
0x0100	CNN
0x0101	Headline News
0x0102	The Weather Channel
0x0103	CNBC
0x0104	CSPAN
0x0105	CSPAN-2
0x0106	America's Talking
0x0107	Talk Channel
0x0108	Court TV
0x0109	The Crime Channel
0x010A	National Empowerment TV

Table III: Broadcast Channel Identification	
0x0120 to	
0x013F	**Sports**
0x0120	ESPN
0x0121	ESPN-2
0x0122	SportSouth
0x0123	The Golf Channel
0x0124	Classic Sports Network
0x0125	Prime Network
0x0126	NewSport
0x0140 to	
0x015F	**Music**
0x0140	MTV
0x0141	VH-1
0x0142	Country Music Television
0x0143	The Nashville

	Network
0x0144	The Box
0x0145	Video Jukebox
0x0146	MOR Music TV
0x0147	Music Choice

[0074I] Table IV below shows some possible identification codes for particular applications. Note that each application could be programmed to insert its application ID code into the event record without accessing table IV. But by having each application access the table IV during the journaling process, the system's ability to modify or add application ID codes easily is enhanced because such codes could be populated across system by downloading an updated table IV. Providing for downloading of new tables increases the application footprint and system complexity so tables can also be part of the application programming.

Table IV: Application Identifiers	
ID Code	**Content**
0x0000	Operating System
0x0001-F	Operating System Sub-Systems
0x0010	Application Manager
0x0011	Cable Television Application
0x0012	Clickstream Kernel
0x0100	EPG System
0x0101	Digital Pictures - Interactive Game
0x0110-F	Viacom - MTV / Showtime, etc.
0x1000	Interplay Written Applications General ID
0x1001	Interplay Runtime Engine
0x1002	Interplay Navigator
0x1003	Interplay VOD

0x1004	Interplay NVOD
0x1005	Interplay TownGuide
0x1100	The Weather Channel, Weather On-Demand
0x1101	Worldspan - Travel On-Demand
0x1102	Lightspan - Educational Interactive Application
0xFFFF	Missed Events Record

[0074J] Each particular application can simply reference the global application, event and channel identification tables (which periodically may be updated and then downloaded to STBs) in order to build an event record. Examples of application specific event records that may be created in this manner are shown in Tables V through VIII below and discussed in associated text.

[0074K] A cable TV application may tune analog or digital broadcast services. When a command to change channels is entered, the cable TV application is invoked. The cable TV application begins building an event record by inserting an application ID and time stamp into the record. Next, the application determines the "event ID" by cross-referencing the command with the global event ID table II for the proper code. Then, the application journals the "Channel ID."

[0074L] Although the Channel ID could simply be the number of the channel, that information means little. The fact that channel 6 was watched more than channel 7 has little or no meaning unless networks and, ultimately, the content delivered by those networks are associated with particular channels. Accordingly, the Channel ID may be a field, like a 16 bit field, which uniquely identifies the broadcast network displayed on that particular channel. The Channel ID may be determined by programming the cable TV application to compare the channel number tuned with global broadcast channel identification table III, above, to determine the correct channel identification code. Correlating the channel number

with the channel identification code found in Table III ensures accurate reporting even though channels may differ at different cable TV headends within a particular region or even though individual channel line-up changes may be made over a period of time. This correlation between channel number and channel identification code could be done also at the staging server after it receives all of the event records, provided that correlation there accounted for different regional channel lineups.

Table V: Cable TV Application Event Record	Size
Application ID: See Application ID table IV	16 bits
Timestamp: Identifies event time	6 bytes
Event ID: See Global Event ID table II for Syntax	16 bits
Channel ID : See Broadcast Channel ID table III for Syntax	16 bits

[0074M] Table VI below shows a navigator application that may be provided in order to give subscribers an interactive menu that assists them in selecting from the many available programs and applications in an interactive network. The "Event ID" refers to the identification codes for commands relating to the Navigator application, which codes may be located by referring to the global event ID table II above. Table VI also shows some of the features of the navigator that might be used by the subscriber and that could be useful to track. The right hand column under "Size/Data" shows, first, next to the "Application state ID" that 8 bits are allocated to that record and, second, in the various rows beneath, the particular code that is journaled in order to indicate a subscriber accessed the identified (e.g. Fly-Thru, Main Menu, etc.) screen. Such information lets system operators determine the screens that users are viewing heavily or lightly in order to replace less popular screens with more useful ones or to charge more for advertisements placed on heavy use screens.

Table VI: Navigator Application Event Record	Size/ Data

Application ID: See Application ID table IV	16 bits
Timestamp: Identifies event time	6 bytes
Event ID: See Global Event ID table for Syntax	16 bits
Application State ID: See below for information tracked:	8 bits
Fly-Thru	0x00
Main Menu	0x01
Information (Help) Screen or Video	0x02
Movies Sub-Menu	0x03
Movie Categories Sub-Menu	0x04
List of Movies Sub-Menu	0x05
Movie Info Screen	0x06
Movie Buy State	0x07

[0074N] Table VII similarly shows the journaling information collected for a video on demand application that may be launched in an interactive service from the Navigator application above or its equivalent. Some of the information collected here may include the amount of pausing, fast forwarding and rewinding. Additionally, the service provider may want to determine whether viewers are recording a video in order to charge them a recording fee. Similar information could be collected for a near video on demand service, which typically allows only incremental pause, forward or rewind.

Table VII: Video on Demand Application Event Record	**Size/ Data**
Application ID: See Application ID table IV	16 bits
Timestamp: Identifies event time	6 bytes
Event ID: See Global Event ID table for Syntax	16 bits
Application State ID: See below for information tracked:	8 bits
Playing	0x00
Paused	0x01

Fast Forward	0x02
Rewind	0x03
Info (Help) Video or Screen Played	0x04
Reserved	0x05
Reserved	0x06
Reserved	0x07

[0074O] Table VIII below shows the event record for the Electronic Program Guide (EPG) application. The EPG application records the application ID, timestamp and event ID records just as do the above applications described in tables V-VII. Additionally, it has an application state ID field that identifies which of the display screens were accessed by subscribers, as shown below.

Table VIII: Electronic Program Guide (EPG) Application Event Record	**Size/ Data**
Application ID: See Application ID table IV	16 bits
Timestamp: Identifies event time	6 bytes
Event ID: See Global Event ID table for Syntax	16 bits
Application State ID: See below for information tracked:	8 bits
Initial Display Screen	0x00
Look Ahead Display 4 Hour	0x01
Look Ahead Display 8 Hour	0x02
Look Ahead Display 12 Hour	0x03
Look Ahead Display 16 Hour	0x04
Look Ahead Display 20 Hour	0x05
Look Ahead Display 24 Hour	0x06
Reserved	0x07

[0074P] Generally, similar information about other applications, such as home shopping, interactive gaming or any other new applications deployed over an

interactive or other media delivery system, can be tracked in a similar fashion. Additionally, the journaling process may be used to track errors within the system, with clickstream kernel journaling such errors using the same method as described above.

[0074Q] Over time, the journaling needs of system, or system itself may evolve. Applications may be changed or new ones deployed. New events may become of interest to the operator of system. In order to provide flexibility for system, operators may download to STBs new or replacement applications that will include the necessary processes for journaling all events of interest.

[0074R] Briefly, the aim of the merge and parse process is to merge each STB event records with various "metadata." "Metadata" refers to (1) programming of virtually any type shown on system including the time and broadcast or cable network providing such programming or (2) interactive applications invoked by subscribers. For instance, metadata includes the following sources of data: EPG broadcast programming schedule data, broadcast advertising schedule data, local advertising schedule data or session-services advertising schedule data and session-services programming schedule data. As used herein, "session-services advertising" refers to advertising inserted by video server (or alternate insertion means) during particular interactive sessions with the subscriber (via the STB) that are the session-services programming.

[0074S] Collectively, all of this data enters into a merge and parse engine that creates an event timeline for each STB. Merge and parse engine may be deployed upon staging server or the MKIS system. So deploying merge and parse engine on staging server allows collected event records to be merged and parsed. The resulting event timelines can be sent to MKIS system for further analysis.

[0074T] Timeline provides a snapshot of activity on a particular STB for a selected period (*e.g.*, 24 hours) or for a selected event -- for instance, a timeline would be created for each STB tuning to a particular show or shows (*e.g.*, a pay per view fight) that may occur over a selected period. Timeline is created by merging event records with metadata about programming available over the network for the selected time period.

[0074U] To merge that data, proper priority must be assigned to data that otherwise may be conflicting. For instance, broadcast advertising data may indicate that a certain national ad was run at Time A. On the other hand, if the system is an interactive system and the interactive server provided a targeted advertisement ("ad") also at Time A, as indicated by session-services advertising data, that targeted ad was inserted over the national ad at Time A. Thus, by assigning session-services advertising data a priority higher than national broadcast advertising data, the merge and parse engine is able to create an accurate timeline of programming delivered to a particular STB. Similarly, even a traditional cable or wireless cable network requires priority assignments. Typically, local cable operators typically are allowed to insert local ads over certain national ads (assuming they can sell that local ad time).

[0074V] Priority assignments may relate to several sources of data, such as EPG metadata, National and Local Insert ad metadata and Interactive Sessions metadata. EPG metadata is usually very broad -- for instance, showing a football game on channel 1 from 1:00 to 4:00 p.m. Thus, EPG metadata is assigned a priority lower than that of national ad metadata because a particular national ad will be overlayed into a particular time slot broadly defined by the EPG. In turn, local insert ad metadata trumps national ad metadata because the national ad metadata may not account for situations where a local network or affiliate inserts a local ad over the national ad scheduled for a particular timeslot. Finally, interactive sessions metadata, which reflects subscriber selections, has the highest

priority as it shows the subscriber stopped watching a particular channel and instead invoked an interactive session.

[0074W] Applying these priority rules produces a timeline for each subscriber. Additional filtering criteria are applied by the merge and parse engine in order to generate a further refined timeline. For example, event records may include such highly granular and specific information as the number of volume ups or channel ups that a particular subscriber entered. One set of filtering criteria may ensure that the timeline includes only channels that were viewed for more than a threshold (*e.g.*, 15 seconds) time period. This eliminates any very fast channel changes made by the subscribers, thereby simplifying the event timeline because event records that do not meet the criteria are filtered out of the event timeline.